Counting Models Using Extension Rules

Minghao Yin, Hai Lin, Jigui Sun

College of Computer Science and Technology, Jilin University, Changchun, China, 130012
mhyin@nenu.edu.cn

Introduction

In recent years we have viewed tremendous improvements in the field of Propositional satisfiability (SAT). Many NP-complete problems from a variety of domains, such as classic planning problems (Kautz 1999), have been compiled into SAT instances, and solved effectively by SAT solvers (Zhang et al 2001). On the other hand, the problem of counting the number of models of a prepositional formula (#SAT) is an important extension of satisfiability testing (Bacchus et al 2003). Recent research has also shown that model counting corresponds to numerous #P-complete problems such as performing inference in Bayesian networks (Sang et al 2005) and conformant probabilistic planning (Domshlak and Hoffman. 2006).

Resolution principle is the rule of inference at the basis of most procedures for both SAT and #SAT, though a number of techniques, such as clause learning, variable selection, can be integrated to improve performance tremendously. The aim of this paper is to challenge the traditional idea by using the inverse rule of resolution, which we called *extension rule* (Lin et al 2003). Specifically, the basic idea is to deduce the set of all the maximum terms for counting models and to use the inclusion-exclusion principle to circumvent the problem of space complexity. Our contributions are:

(1) We use the inverse of resolution together with the inclusion-exclusion principle for counting models. This may be considered a novel framework for model counting.
(2) Our method leads to a new target logical language, which permits model counting to be solved in linear time.
(3) We propose a revised method to lift our method from model counting to weighted model counting (WMC).

Extension Rule

We will begin by specifying the notation that will be used in the rest of this paper. We use Σ to denote a set of clauses in Conjunctive Normal Form (CNF), C to denote a single clause, and M to denote the set of all atoms that appear in Σ. The Extension rule is defined as follows:

Definition 1 Given a clause C and a set M: C'={C∨a, C∨¬a | "a" is an atom, a∈M and "a" doesn't appear in C}. We call the operation proceeding from C to C' the *Extension rule* on C and C' the result of the extension rule.

Definition 2 A clause is a maximum term on a set M iff it contains all atoms in M in either positive or negative form.

Model Counting via Extension Rule

The following theorem is used to count the numbers of models of a given set of maximum terms.

Theorem 1 Given a set of clauses Σ with its set of atoms M (|M|=m), if the clauses in Σ are all maximum terms on M, if Σ contains S distinct clauses, then the number of models of Σ is 2^m-S.

According to theorem 1, if we want to count models of a set of clauses, we should proceed by finding an equivalent set of clauses so that all the clauses in it are maximum terms by using extension rule; then using theorem 1, We can know how many clauses there are. We call this process *model counting based on extension rule.* We have demonstrated that model counting based on extensionrule is sound and complete in (Yin et al 2006).

However, it is clear that a direct use of the Extension rule (generate all the maximum terms) is infeasible due to considerations of space complexity. The key is that it is sufficient to *count* the clauses rather than to *list* them. We have showed in (Lin et al, 2003) how to circumvent the space by using the inclusion-exclusion principle.

We can count the maximum terms as follows: Given a set of clauses Σ={C_1, C_2, …,C_n}, let M be the set of atoms which appear in Σ (|M|=m). Let P_i be the set of all the maximum terms we can get from C_i by using the Extension rule, and let S be the number of distinct maximum terms we can get from Σ. By using the famous inclusion-extension principle, we will have equation (1):

$$S = |P_1 \cup P2 \cup \ldots \cup P_n| = \sum_i |P_i| + \sum_{1\le i<j\le n} |P_i \cap P_j| - \sum_{1\le i<j<l\le n} |P_i \cap P_j \cap P_l| + (-1)^{n+1} |P_1 \cap P_2 \cap \ldots \cap P_n|$$

where, $|P_i| = 2^{m-|C_i|}$ ·

$$|P_i \cap P_j| = \begin{cases} 0, \text{There are complementary literals in } C_i \cup C_j. \\ 2^{m-|C_i \cup C_j|}, \text{ Otherwise.} \end{cases}$$

Using equation (1), we can easily design an algorithm for exact model counting.

Model Counting via Knowledge Compilation

In (Lin et al 2004), we have defined a new tractable class as *EPCCL* (each pair of clauses contains complementary literal(s)) theory and show that it is qualified as a target language for knowledge compilation. We also showed how to find an equivalent EPCCL theory of the original theory using extension rule. Here we show that EPCCL theory is qualified as a target language for counting models in polynomial time.

Definition 3 An EPCCL theory is a set of clauses in which each pair (of clauses) contains complementary literal(s).

Given an EPCCL theory, counting its models can be done in linear time using the extension rule. The reason is simple: only the first n (the number of clauses) terms need to be computed in equation (1), all the other terms being zero. We have proved in (Yin et al, 2006) that counting models about an EPCCL theory can be done in polynomial time. For the compilation process to find an equivalent EPCCL theory of the original theory using extension rule, we refer to (Yin et al 2006).

WMC via Knowledge Compilation

Given a logical theory Σ, and for each literal l, it is assigned a weight W(l). The weights for the literals induce a weight for each model ω of Σ as follows:

$$W(\omega) = \prod_{\omega |= l} W(l)$$

Computing WMC for Σ, which we denote WMC(Σ), is computing the sum of the weights of all models of Σ:

$$WMC(\Sigma) = \sum_{\omega |= \Sigma} W(\Sigma)$$

Now we show given an EPCCL theory Σ, how to compute WMC(Σ).

Remark 1. Given a theory Σ, let M be the set of atoms which appear in Σ, M = $\{l_1,\ldots, l_m\}$ if Σ|=T, then

$$WMC(\Sigma) = \prod_{i=1}^{m} [w(l_i) + w(\neg l_i)]$$

Remark 2. Let Σ be a logical theory and M be the set of atoms appearing in Σ, M = $\{l_1,\ldots, l_m\}$, $l_k \in M$, $1 \le k \le m$, then

$$WMC(\Sigma(l_k)) = w(l_k) * \prod_{i=1}^{k} [w(l_i) + w(\neg l_i)] * \prod_{i=k+1}^{m} [w(l_i) + w(\neg l_i)]$$

Here $\Sigma(l_k)$ means that we restrict l_k to 1 when finding models in Σ. Similarly, we can compute $\Sigma(\neg l_k)$ easily.

The following theorems are used to compute the weights of all models of an EPCCL theory Σ.

Theorem 2. Given an EPCCL theory Σ = $\{C_1, C_2, \ldots, C_n\}$, and let C_i be any clause in Σ if an assignment makes C_i false, then it will make other clause in Σ true.

Theorem 3. Given an EPCCL theory Σ contains n clauses: $\Sigma = \{(l_{11} \vee \ldots \vee l_{1n_1}), \ldots, (l_{n1} \vee \ldots \vee l_{1n_n})\}$. Let the set of atoms appearing in Σ denoted by M = $\{l_1,\ldots, l_m\}$, then

$$WMC(\Sigma) = \prod_{i=1}^{m} [w(l_i) + w(\neg l_i)] - WMC(\neg l_{11} \wedge \ldots \wedge \neg l_{1n_1}) - \ldots - WMC(\neg l_{n1} \wedge \ldots \wedge \neg l_{nn_n})$$

Example Let Σ = {p∨q, p∨¬q, ¬p∨r}; w(p)=w(¬p)=0.5; w(q) = 0.3, w(¬q)=0.7; w(r) = 0.8, w(¬r) = 0.2. Then the weights for all possible assignments are:

WMC(T) = (0.5+0.5)*(0.3+0.7)*(0.2+0.8) = 1

All the possible assignments make p∨q to be false must have p=0 and q=0, so

WMC(Σ(¬p∧¬q)) = 0.5*0.7*(0.2+0.8) = 0.35;

Similarly, WMC(Σ(¬p∧q)) = 0.5*0.3*(0.2+0.8) = 0.15;

WMC(Σ(p∧¬r)) = 0.5*(0.3+0.7)*0.2 = 0.1.

WMC(Σ) = 1 – 0.35 – 0.15 – 0.1 = 0.4.

Conclusion

This paper explores what happens if we use the inverse of resolution for model counting and weighted model counting. We can compare our method with resolution-based method. The more pairs of clauses with complementary literals, the more efficient our method is, and the less efficient resolution-based method is. In this sense, our method can be considered as a counterpart to resolution-based method for model counting and weighted model counting. The paper is a first attempt to do this work, and future work must be done in order to obtain more refined work.

Acknowledgement

This work was fully supported by Specialized Research Fund for the Doctoral Program of Higher Education (Grant No.20050183065).

References

F. Bacchus, etc. 2003. Algorithms and Complexity Results for #SAT and Bayesian inference. *In FOCS*'03.

R. J. Bayardo Jr. and J. D. Pehoushek 2000, Counting models using connected components, *In AAAI'00*.

E. Birnbaum and E. L. Lozinskii 1999. The good old DP procedure helps counting models. *JAIR*, 10: 457-477.

M. Chavira, A. Darwiche 2005. Compiling Bayesian networks with local structure. *In IJCAI'05*.

C. Domshlak, J. Hoffmann, 2006. Fast probabilistic planning through weighted model counting, *In ICAPS'06*.

H. Kautz and B. Selman 1999. Unifying sat-based and graph-based planning. In *IJCAI'99*.

H. Lin, J. Sun, Y. Zhang 2003. Theorem proving based on extension rule. *JAR, 31(1), 11-21*.

H. Lin, J. Sun 2004. Knowledge compilation using extension rule. *JAR,* 32(2), 93-102.

T. Sang, etc. 2004. Combining component caching and clause learning for effective model counting. *In SAT'04*.

T. Sang, P. Beame, and H. Kautz 2005. Solving baysian networks by weighted model counting. *In AAAI'05*.

M. Yin, etc. 2006, extension or resolution: a novel approach for counting models, Tech. R. of Jilin University.

L. Zhang, etc. 2001. Efficient conflict driven learning in a Boolean satisfiability solver. In ACM/IEEE *ICCAD'01*.